UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/614,664	07/07/2003	Dikran Meliksetian	POU920020113US1	7571

7590 02/22/2006

John E. Campbell
IBM Corporation
2455 South Road, P386
Poughkeepsie, NY 12601

EXAMINER	
LE, MICHAEL	
ART UNIT	PAPER NUMBER
2163	

DATE MAILED: 02/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/614,664	Applicant(s) MELIKSETIAN ET AL.	
	Examiner Michael Le	Art Unit 2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 July 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 July 2003* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Summary and Status of Claims

1. This Office Action is in response to Applicant's reply filed July 7, 2003.

2. Claims 1-30 are pending.

3. The drawings are objected to.

4. The specification is objected to for minor informalities.

5. Claim 16 is objected to for minor informalities.

6. Claims 7-9 and 19 are rejected under 35 U.S.C. 112, second paragraph.

7. Claims 29 and 30 are rejected under 35 U.S.C. 101.

8. Claims 1-4, 8, 10, 13, 14, 17, 19, 20 and 27-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371), in view of Fernandez et al. (US Patent 6,604,100).

9. Claims 5-7, 9, 11, 12, 15, 16, 18, 22, 23, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371), in view of Fernandez et al. (US Patent 6,604,100), further in view of Fox et al. (US Patent Pub 2003/0120665).

10. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371), in view of Fernandez et al. (US Patent 6,604,100), further in view of Kerwin (US Patent Pub 2003/0212660).

11. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371), in view of Fernandez et al. (US Patent 6,604,100), further in view of Fox et al. (US Patent Pub 2003/0120665), further in view of Kerwin (US Patent Pub 2003/0212660).

Priority

12. No claim of priority has been made. Consequently, claims 1-30 have been examined with a priority date of July 7, 2003.

Drawings

13. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because in **Figure 3C**, reference characters "326" and "327" have both been used to designate a network. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

14. **Figures 1 and 2** should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not

accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

15. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: **Figure 1**, reference characters 108 and 112; **Figure 2**, reference characters 201-205, 210 and 211; **Figure 3C**, reference character 326; **Figure 4**, reference characters 420-427; **Figure 6**, reference character 604; **Figure 7**, reference character 706; **Figure 8B**, reference character 854; **Figure 10A**, reference character 1051; **Figure 10B**, reference characters 1051 and 1057; **Figure 10C**, reference characters 1057 and 1082. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

16. The disclosure is objected to because of the following informalities:

17. In paragraph 0024, line 5, "101" has to be changed according to the figure it references.

18. In paragraph 0025, line 10, “101” has to be changed according to the figure it references. Appropriate correction is required.

19. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

20. **Claims 16 and 22 are objected to** because of the following informalities:

21. In **claim 16**, lines 3-4, it recites “a first run-time object which first run-time object uses”. The second “first run-time object” has to be deleted.

22. In **claim 22**, lines 3, 6, and 9, “second run-time object” is recited, however there is no first run-time object in the claim tree. Accordingly, “second” in each occurrence has to be changed to --first--.

23. Appropriate correction is required.

Claim Rejections - 35 USC § 112

24. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

25. **Claims 7-9 and 19 are rejected under 35 U.S.C. 112, second paragraph**, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

26. **Claim 7** recites the limitation "the second run-time object" in lines 4 and 6. There is insufficient antecedent basis for this limitation in the claim. For the prior art rejections, the Examiner will interpret the "second run-time object" as the --first run-time object--.

27. **Claim 9** recites the limitation "the XSL transformation" in line 3. There is insufficient antecedent basis for this limitation in the claim. For the prior art rejections, the Examiner will interpret "the XSL transformation" as --an XSL transformation--.

28. **Claim 19** uses the slash symbol in line 3 between "update" and "insert". It is unclear whether the slash symbol means "and" or "or". For the prior art rejections, the Examiner interprets the slash symbol as meaning --or--.

29. **Claim 8** is rejected because it depends on a rejected claim. Dependent claims contain the limitations of the parent claims and are therefore rejected for the same reasons.

30. The prior art rejections below for claims 7-9 and 19 are made as best understood in light of the 35 U.S.C. 112, second paragraph rejections addressed above.

Claim Rejections - 35 USC § 101

31. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 29 and 30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The basis of this rejection is set forth in a test of whether the invention is categorized as a process, machine, manufacture or composition of matter and if the invention produces a useful, concrete and tangible result. Mere ideas in the abstract (i.e., abstract idea, law of nature, natural phenomena) are found to be non-statutory subject matter. For a method claim to pass muster, the recited process must produce a useful, concrete and tangible result.

In the present case, **claims 29 and 30** recite a system, however the components of the system are merely software per se. A system claim must recite physical structure thus enabling it to be properly categorized in one of the statutory categories of invention. Since the components of the systems claims in claims 29 and 30 are software per se and do not contain any physical components, the systems can not be categorized in one of the statutory categories of invention and is thus nonstatutory.

To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. 101 (nonstatutory) above are further rejected as set forth below in anticipation of applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 103

32. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

33. Claims 1-4, 8, 10, 13, 14, 17, 19, 20 and 27-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) hereinafter "Fernandez".

34. In regards to **claims 1,** Dodds discloses a computer program product for creating an XML representation of data stored in a relational database system (Dodds: para. 0007), the computer

program product comprising a computer readable medium having computer readable program code therein (Dodds: para. 0020), the computer program product comprising:

a. computer readable program code for specifying a set of conditions that data to be retrieved from the relational database system must satisfy (Dodds: para. 0041, lines 3-5)[1];

b. computer readable program code for specifying an output format that the XML representation must satisfy (Dodds: para. 0041, lines 7-11, 13-18)[2];

c. computer readable program code for retrieving data from the relational database using a standard database access method (Dodds: para. 0042-0045)[3]; and

35. Dodds does not expressly disclose creating from the set of conditions and the format, a mapping description being in a markup language comprising XSL and SQL function, retrieving data using the mapping description and formatting the XML object representing the retrieved data using the mapping description. Dodds does disclose outputting an XML object representing the retrieved data (Dodds: para. 0041, lines 7-11, 13-18). Dodds further discloses in the background information that XSL is used to format XML data to be displayed (Dodds: para. 0003, lines 11-14).

36. Fernandez discloses a method for converting relational data into a structured document (Fernandez: col. 2, lines 27-29). Fernandez further discloses an RXL language for generating a query to retrieve relational data and transform it into an XML view (Fernandez: col. 5, lines 4-8). Fernandez also discloses composing an executable query, which is comprised of a data-

[1] A user query is interpreted as a set of conditions. It is a query to a relational database, so it is specifying conditions for data to be retrieved from the relational database system.

[2] The XML is reconstructed according to the sections (format) that are retrieved, which are specified by the user's query.

[3] SQL is used (standard database access method) to retrieve the data from the database.

extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45).

37. Dodds and Fernandez are analogous art because they are from the same field of endeavor of mapping relational data to XML.

38. At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the computer program product of Dodds by adding computer readable code for creating from the set of conditions and the format, a mapping description (execution query taught by Fernandez) being in a markup language comprising SQL and XSL function (disclosed by Dodds in the background), retrieving data using the mapping description and formatting the XML object representing the retrieved data using the mapping description, as taught by Fernandez.

39. The motivation for doing so would have been because it is desirable to devise an efficient means of storing, indexing and retrieving via queries XML documents in a relational database system (Dodds: para. 0004, lines 1-4). Fernandez discloses an efficient tool for querying relational data in XML, which meets the needs of Dodds (Fernandez: col. 2, lines 39-33).

40. In regards to **claim 13**, Dodds discloses updating information stored in a relational database based on data extracted from an XML file (Dodds: para. 0024, lines 1-4). Dodds also discloses retrieving data from a relational database and outputting the data as XML as addressed in the rejection to claim 1 above. Claim 13 seems to be the opposite of claim 1. Also addressed

above, Fernandez discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). The executable query is interpreted as the mapping description. Accordingly, claim 13 is rejected because it contains the same elements as claim 1 and at the time of the invention, it would have been obvious to one of ordinary skill in the art to do the opposite of claim 1, to store the XML documents in a relational database. The motivation for doing so would have been because it is desirable to retrieve data that is stored.

41. In regards to **claim 2**, the limitation was addressed above in the rejection to claim 1. Fernandez also discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries (first section) and an XML-construction part, such as an XML template (second section) (Fernandez: col. 5, lines 26-30).

42. In regards to **claim 3**, the limitation was addressed above in the rejection to claim 1. Fernandez also discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries (first section) and an XML-construction part, such as an XML template (second section) (Fernandez: col. 5, lines 26-30). One or more SQL queries is interpreted as a list of individual search conditions, wherein each condition is a SQL query.

43. In regards to **claim 4**, the limitation was addressed above in the rejection to claim 1. Fernandez also discloses composing an executable query, which is comprised of a data-

extraction part of one or more SQL queries (first section) and an XML-construction part, such as an XML template (second section) (Fernandez: col. 5, lines 26-30). The SQL queries in the execution query specify one or more tables upon which the statement will act (Fernandez: col. 9, line 27; col. 10, line 9), one or more columns of the table that will be retrieved (Fernandez: col. 10, line 8), a list of parameters for determining a location where a value of a parameter is to be obtained (Fernandez: col. 9, lines 28-29; col. 10, lines 10-12) and the value of the parameter to be used in a condition clause of the SQL prepared statement (Fernandez: col. 9, lines 28-29; col. 10, lines 10-12).

44. In regards to **claim 8**, Dodds discloses generating software events (Dodds: para. 0024, line 6)[4] and generating data associated with the software events (Dodds: para. 0024, lines 7-9)[5].

45. Dodds does not expressly disclose sequencing the software events and the associated data based on the internal representation of the retrieved data.

46. Fernandez discloses a method for converting relational data into a structured document (Fernandez: col. 2, lines 27-29). Fernandez further discloses an RXL language for generating a query to retrieve relational data and transform it into an XML view (internal representation) (Fernandez: col. 5, lines 4-10).

47. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the computer program product of Dodds by adding computer readable program code for sequencing the software events and the associated data based on the internal representation of the retrieved data, taught by Fernandez.

[4] The database processes the query (the database processing the query is a software event).

48. The motivation for doing so would have been because it is desirable to devise an efficient means of storing, indexing and retrieving via queries XML documents in a relational database system (Dodds: para. 0004, lines 1-4). Fernandez discloses an efficient tool for querying relational data in XML, which meets the needs of Dodds (Fernandez: col. 2, lines 39-33).

49. In regards to **claim 10**, Dodds discloses in the background information that XSL is commonly used for formatting XML documents (Dodds: para. 0003, lines 11-14).

50. Dodds does not expressly discloses an internal representation of the data retrieved from the database according to the specification that is formatted to be output.

51. Fernandez discloses an internal representation of the data retrieved from the database according to the specification that is formatted to be output (Fernandez: col. 5, lines 38-45).

52. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the computer program product of Dodds by adding computer readable code for applying XSL syntax to transform an internal representation of the data retrieved from the database according to the specification that is formatted to be output, taught by Fernandez.

53. The motivation for doing so would have been because it is desirable to devise an efficient means of storing, indexing and retrieving via queries XML documents in a relational database system (Dodds: para. 0004, lines 1-4). Fernandez discloses an efficient tool for querying relational data in XML, which meets the needs of Dodds (Fernandez: col. 2, lines 39-33).

[5] The query results are generated (data associated with the processing (software event)).

54. In regards to **claim 14**, the mapping description was addressed above in the rejection to claim 13 as being disclosed by Fernandez. Fernandez discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). The executable query is interpreted as the mapping description. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the XML-construction part to be an XML retrieval part for extracting data from the XML file. The motivation for doing so would have been because it is desirable to retrieve data that is stored.

55. In regards to **claim 17**, the limitation was addressed above in the rejection to claim 13. Fernandez discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). The executable query is interpreted as the mapping description. The data-extraction part contains one or more SQL queries (sequence of database operations and their respective inputs).

56. In regards to **claim 19**, Dodds discloses database operations for storing (insert) XML data in a relational database (Dodds: para. 0036, lines 7-8).

57. In regards to **claim 20**, Dodds discloses database operations that are SQL prepared statements (Dodds: para. 0043).

58. **Claim 27** is substantially similar to claim 1 in the form of a method and is rejected for the same reasons.

59. **Claim 28** is substantially similar to claim 13 in the form of a method and is rejected for the same reasons.

60. **Claim 29** is substantially similar to claim 1 in the form of a system and is rejected for the same reasons.

61. **Claim 30** is substantially similar to claim 13 in the form of a system and is rejected for the same reasons.

62. **Claims 5-7, 9, 11, 12, 15, 16, 18, 22, 23, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) hereinafter "Fernandez", further in view of Fox et al. (US Patent Pub 2003/0120665) hereinafter "Fox".**

63. In regards to **claim 5**, Dodds and Fernandez do not expressly disclose creating a first runtime object representing elements of the mapping description wherein the run-time object represents a search condition as a JDBC prepared statement and defines Java variables for the values of the parameters.

64. Fox discloses a method of transforming data from one data schema to another data schema (Fox: para. 0103), wherein the source and target schemas are analyzed and mapped, then

transformed based on the mappings (Fox: para. 0107). Fox further discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

65. Dodds, Fernandez and Fox are analogous art because they are all directed towards the same field of endeavor of data transformation.

66. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for creating a first runtime object representing elements of the mapping description wherein the run-time object represents a search condition as a JDBC prepared statement and defines Java variables for the values of the parameters, as taught by Fox.

67. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

68. In regards to **claim 6**, Dodds and Fernandez disclose producing an internal representation of the data retrieved from the database (Fernandez: col. 5, lines 4-8), providing a value for a variable needed for a first SQL prepared statement (Fernandez: col. 6, lines 29-34), executing the first SQL prepared statement (Fernandez: col. 5, lines 4-8) and storing a result of the executing step as an internal representation (Fernandez: col. 5, lines 8-10).

69. Dodds and Fernandez do not expressly disclose that the first run-time object produces the internal representation.

70. Fox discloses a method of transforming data from one data schema to another data schema (Fox: para. 0103), wherein the source and target schemas are analyzed and mapped, then transformed based on the mappings (Fox: para. 0107). Fox further discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7).

71. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for the first run-time object to produce the internal representation, as taught by Fox.

72. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

73. In regards to **claim 7**, Dodds and Fernandez do not expressly disclose invoking an XSL transformation engine and operating on the internal representation with the first run-time object.

74. Fox discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

75. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for invoking an XSL transformation engine and operating on the internal representation with the first run-time object, taught by Fox.

76. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

77. In regards to **claim 9**, Dodds and Fernandez do not expressly disclose providing the software events and associated data to an XSL transformation, the providing being performed for a single event. Dodds does disclose in the background information that XSL is used for formatting XML (Dodds: para. 0003, lines 11-14).

78. Fox discloses XSLT transformations wherein an XSLT script (software events and associated data) is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed (single event) by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

79. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for providing the software events and associated data to an XSL transformation, the providing being performed for a single event, taught by Fox.

80. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

81. In regards to **claim 11**, Dodds and Fernandez do not expressly disclose creating a second run-time object using a Xalan Templates object for representing the XSL transformation.

82. Fox discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

83. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for creating a second run-time object using a Xalan Templates object for representing the XSL transformation, taught by Fox.

84. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications. Furthermore, Xalan is known XSL Transformation processor which is commonly used with Java as it is implemented as a Java library.

85. In regards to **claim 12**, Dodds and Fernandez do not expressly disclose creating a run-time object, saving the run-time object in a run-time object cache and using the cached run-time object.

86. Fox discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox further discloses run-time transformations (run-time objects) that are created (Fox: para. 0164, lines 1-2), saved to a cache (run-time cache) (Fox: para. 0164, lines 3-8) and used from the cache (Fox: para. 0164, lines 7-8).

87. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for creating a run-time object, saving the run-time object in a run-time object cache and using the cached run-time object, as taught by Fox.

88. The motivation for doing so would have been because it saves times because the object need not be generated again (Fox: para. 0166, lines 10-16).

89. In regards to **claim 15**, Dodds discloses retrieving XML object data (Dodds: para. 0024, lines 1-4). Fernandez discloses creating an internal representation of retrieved data (Fernandez: col. 5, lines 4-10).

90. Dodds and Fernandez do not expressly disclose specifying in XSL syntax a location of the XML object data to be retrieved from the XML object, wherein the XSL syntax determines a transformation and creating the internal representation of required data from the XML object data based on the transformation.

91. Fox discloses a method of transforming data from one data schema to another data schema (Fox: para. 0103), wherein the source and target schemas are analyzed and mapped, then transformed based on the mappings (Fox: para. 0107). Fox further discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

92. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable program code for specifying in XSL syntax a location of the XML object data to be retrieved from the XML object, wherein the XSL syntax determines a transformation and creating the internal representation, taught by Fernandez, of required data from the XML object data based on the transformation, taught by Fox.

93. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

94. In regards to **claims 16, 18, 22 and 25**, the limitations were addressed above in the rejection to claims 13 and 17 as being disclosed by Fox. Fox discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153). Fox discloses using Xalan (claim 16), XSL scripts (claim 18), Java run-time objects using JDBC prepared statements and Java variables (claim 22) and the process of creating the run-time object and running an XSL script using an XSL processing engine (claim 25).

95. In regards to **claim 23**, Dodds and Fernandez disclose associating with an SQL statement, a value of a variable from the internal representation of retrieved data and executing the SQL statement (Fernandez: col. 5, lines 13-21).

96. **Claim 26** is essentially claim 12 and is rejected for the same reasons.

97. **Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) hereinafter "Fernandez", further in view of Kerwin (US Patent Pub 2003/0212660).**

98. Dodds and Fernandez do not expressly disclose sequencing database operations as segmented transactional groups.

99. Kerwin discloses a transaction server that provides sequenced SQL statements (Kerwin: para. 0075). Kerwin further discloses queued SQL statements (transactional groups) (Kerwin: para. 0179).

100. Dodds, Fernandez and Kerwin are analogous art because they are all directed towards the same field of endeavor of data management using databases and XML.

101. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable program code for sequencing database operations as segmented transactional groups, taught by Kerwin.

102. The motivation for doing so would have been because it is desirable to ensure that commands occur in a proper sequence thereby giving an expected result (Kerwin: para. 0166).

103. **Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) hereinafter "Fernandez", further in view of Fox et al. (US Patent Pub 2003/0120665) hereinafter "Fox", further in view of Kerwin (US Patent Pub 2003/0212660).**

104. Dodds and Fernandez do not expressly disclose database operations that are performed by respecting predefined transactional boundaries such that an operation in a group is only completed if all the operations of the group can be executed successfully.

105. Kerwin discloses a transaction server that provides sequenced SQL statements (Kerwin: para. 0075). Kerwin further discloses queued SQL statements (transactional groups) (Kerwin: para. 0179). Kerwin also discloses that the next statement in a queue is not executed until receiving confirmation of successful execution of the currently executed statement (Kerwin: para. 0175).

106. Dodds, Fernandez and Kerwin are analogous art because they are all directed towards the same field of endeavor of data management using databases and XML.

107. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable program code for performing database operations by respecting predefined transactional boundaries such that an operation in a group is only completed if all the operations of the group can be executed successfully, taught by Kerwin.

108. The motivation for doing so would have been because it is desirable to ensure that commands occur in a proper sequence thereby giving an expected result (Kerwin: para. 0166).

Conclusion

109. The following are prior art made of record and not relied upon but is considered pertinent to applicant's disclosure.

110. Bhatt et al. (US Patent Pub 2003/0101169) discloses a relational database system storing XML documents and providing XML query support. Draper et al. (US Patent 6,581,062) discloses a method for storing semi-structured data in a structured manner by creating mapping descriptions in the form of meta-tables. Wotring et al. (US Patent Pub 2002/0010700) discloses

a method for sharing data between relational and hierarchical databases. Murthy et al. (US Patent Pub 2003/0140308) discloses a method for mapping XML schemas to object-relational database systems. Vaschillo et al. (US Patent Pub 2003/0236859) discloses a method for providing an API interface between XML and SQL environments. Rothschiller et al. (US Patent Pub 2004/0172590) discloses a method for converting schema-based hierarchical data into flat data structures. Zhou et al. (US Patent Pub 2004/0220954) discloses a method of translating data from a hierarchical data structure to a relational data structure using a mapping script. Cheng et al. (US Patent 6,366,934) discloses a method for querying structured documents using a database extender. Chang et al. (US patent 6,584,459) discloses a database extender for storing, querying and retrieving structured documents in a relational database. O'Neil et al. (US Patent 6,889,226) discloses a system and method for relational representations of hierarchical data. Jacinto et al. ("Bidirectional Conversion between XML documents and Relational Databases", Sept. 2002) discloses a method of converting XML documents to be stored in relational databases and retrieving them. Fong et al. ("Converting Relational Database Into XML Document", 2001) discloses a method of converting relational database data into an XML document.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Le whose telephone number is 571-272-7970. The examiner can normally be reached on Mon-Thurs : 9:30am-6pm, Fri: 8am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic can be reached on 571-272-4023. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Le
Art Unit 2163
February 15, 2006

UYEN LE
PRIMARY EXAMINER